Singularities in parametric meshing

R. Aubry[1], K. Karamete[1], E. Mestreau[1], and S. Dey[2]

[1] Sotera Defense Solutions, 1501 Farm Credit Drive, Suite 2300, McLean, VA 22102-5011
[2] US Naval Research Laboratory, code 7130, 4555 Overlook Ave SW, Washington DC 20375, USA

Summary. A parametric meshing technique is presented with special emphasis to singularities in the parametric mapping. Singularities are locations where the parametric mapping is highly distorted or even singular. In a NURBS context, this arises when control points are clustered into the same location in three dimensions. Limitations of parametric plane meshing in this context are highlighted, and zero- and first-order surface approximations are commented. In the context of the DOD CREATE-MG project, different CAD kernels and mesh generators communicate as plugins through application programming interfaces (API) . The parametric mesh generator is coupled to the CAD through the Capstone APIs and is independent of a particular CAD kernel. Some CAD kernels do not allow these geometrical constructions while some tolerate it. It is therefore a necessity to handle these degenerate cases properly. Examples illustrate the method 's capabilities.

Keywords: Parametric mesher, singular mapping, degeneracies, surface triangulation, advancing front method.

1 Introduction

It is often argued that two-dimensional parametric meshers are more robust and faster than their three dimensional counterparts. In a two dimensional parametric mesher, the three dimensional informations are brought back to two dimensions through a metric. Therefore, the three dimensional length computation is traded for a metric evaluation, which represents roughly the same computational effort. However, we have met various difficulties with this approach compared to a three dimensional vision, and propose therefore to rely on the three dimensional information as much as possible.

Regarding parametric meshing in the literature, two main categories can be identified, whether the information is purely two dimensional or not. In the first category, Peraire *et al.* [21] propose a parametric approach with an anisotropic advancing front where stretching is taken into account through coordinate rotations aligned with the stretching interpolated from the background grid. A regular element is then generated in that space and mapped back to the parametric plane thereafter. Guan *et al.* [10] extend the advancing front technique to take into account parametric surfaces through a point and edge shift operator to locally simulate the three dimensional proximity in the two dimensional space. Cuillière [8] also mentions briefly the difficulty associated with closed surfaces. An advancing front technique is also proposed that takes into account the metric of the first fundamental form. Reliance on three dimensional information is not reported. The INRIA gamma project has proposed an original approach for parametric surfaces that rely on an anisotropic Delaunay insertion [5]. In [7], the anisotropic Delaunay kernel is coupled with an advancing front point placement. The notion of geometric mesh is emphasized in [12]. Finally, [6] try to remove the strong constrains introduced by the geometry in case of small geometry entitites. In Lee [13], a pure two-dimensional anisotropic advancing front is used without any reference to the three dimensional space.

In the second category, [24] proposes an anisotropic advancing front method in the parametric plane. However, the information is not purely planar as a three dimensional size is stored and used for local queries, complementing the two-dimensional metric. Angles are evaluated in the three dimensional space. The computation of the optimal point in the Riemanian space is also provided without relying on the spectral decomposition of the metric and relies on the first fundamental form. The front strategy relies on sorting the front edges with respect to their three dimensional size. In [23], geometry is represented through bicubic Bezier patches. Singularities in the parametrization are tackled through evaluation in the vicinity of purely singular points, where the tangent plane is not well defined. An advancing front technique is performed only in the parametric space. However, some part of the optimization process takes place in the three dimensional space.

As far as singularities in the parametric plane are considered, the literature is rather scarce. In Rypl *et al.* [23], the case of repeated control points has been considered on the boundary, where multiplicity appears either along one direction due to a bad parametrization, or in both directions at the pole of an octant of a sphere. As the main assumption is to consider singularities at the boundary of the patch, an interpolation from the boundary towards the inside of the surface is performed to recover meaningful informations. In Lee [13], a secondary mapping is considered along a degenerated line. A composition of both mappings is taken into account in the metric field. Therefore, line singularity has been moved to point singularity where a treatment such as reference [23] may be considered. In Lee *et al.* [14], offset layers of elements are added around line degeneracies. However, no details are given to detect these line singularities in a general setting. Singularities are also expected on the boundaries of the patch.

In the context of the DOD CREATE-MG project, different CAD kernels and meshers communicate as plugins through application programming interfaces (API). The mesh generator is therefore independent of a particular kernel as long as these APIs are implemented for a given kernel. Some CAD kernels rely entirely on NURBS engines. In a bottom-up geometrical construction, vertices, edges and surfaces are constructed explicitly as opposed to simple shapes being modified through geometrical operations such as booleans, lofting or sweeping. Therefore, control points, weights and knots may be precisely chosen to control the shape of the geometry. Sometimes, it might be of interest to lump the control vertices of the NURBS surface definition in order to create cones, or three or less sided surfaces, as shown in this work. It is therefore a necessity to be able to handle these cases properly.

Regarding the organisation of the present work, Section 2 recalls the basics of parametric surfaces. Section 3 recalls the method proposed in the present work. Section 4 concentrates on the the treatment applied to singularities. Finally, Section 5 illustrates the capabilities of the method.

2 Parametric surfaces

In this section, parametric surfaces are reviewed, and necessary results are reminded succinctly in order to prepare to the practical application of the parametric mesh generators.

Let $\varSigma \in \mathbb{R}^3$ be a parametric surface, representing the image of a domain $\varOmega \in \mathbb{R}^2$ as:

$$\begin{aligned} \sigma : \varOmega \in \mathbb{R}^2 &\rightarrow \varSigma \in \mathbb{R}^3 \\ (u,v) &\rightarrow \sigma(u,v) \end{aligned} \tag{1}$$

In a surface mesh generation context, two classical requirements would be to respect a size provided in the three dimensional space, and possibly at the same time, to refine areas where curvature is high to capture accurately the geometry. The way to transfer a three dimensional information to the parametric plane is to rely on the first fundamental form of the surface. Similarly, the way to capture the surface variation is to rely on the second fundamental form of the surface as described now.

For the first case, the first fundamental form of the surface [11, 4] reads:

$$I(du, dv) = Edu^2 + 2F\,du\,dv + Gdv^2 \tag{2}$$

Given the tangent plane vectors (τ_u, τ_v), the previous coefficients read:

$$E = \tau_u \cdot \tau_u \tag{3}$$

$$F = \tau_u \cdot \tau_v \tag{4}$$

$$G = \tau_v \cdot \tau_v \tag{5}$$

The first fundamental form relates how distances in the three dimensional space are perceived from the two dimensional space [9]. For the second requirement, principal curvature evaluation is necessary through the Weingarten map, which relies on the first and second fundamental form. The second fundamental form is given by:

$$II(du, dv) = Ldu^2 + 2M\,du\,dv + Ndv^2 \tag{6}$$

Given the normal vector $\mathbf{n}$, the previous coefficients read:

$$L = -\tau_u \cdot \mathbf{n}_u \tag{7}$$

$$M = -\tau_u \cdot \mathbf{n}_u - \tau_v \cdot \mathbf{n}_v \tag{8}$$

$$N = -\tau_v \cdot \mathbf{n}_v \tag{9}$$

Finally, the principal curvature radii and principal directions are the eigenvalues and eigenvectors of the Weingarten map:

$$-W = \begin{pmatrix} L & M \\ M & N \end{pmatrix} \begin{pmatrix} E & F \\ F & G \end{pmatrix}^{-1} \tag{10}$$

Therefore, the relevant metrics in the parametric plane are given in the first case, where the mesh wants to obey a given size distribution by:

$$M_1 = \begin{pmatrix} \tau_u^T \\ \tau_v^T \end{pmatrix} M_{3D} \begin{pmatrix} \tau_u & \tau_v \end{pmatrix} \tag{11}$$

where M_{3D} is a provided metric in the three dimensional space, and in the second case, where a geometry approximation is sought, by:

$$M_2 = \begin{pmatrix} \tau_u^T \\ \tau_v^T \end{pmatrix} \begin{pmatrix} v_1 & v_2 & n \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \alpha \end{pmatrix} \begin{pmatrix} v_1^T \\ v_2^T \\ n^T \end{pmatrix} \begin{pmatrix} \tau_u & \tau_v \end{pmatrix} \tag{12}$$

where λ_i are the principal eigenvalues, v_i are the principal eigenvectors and α an arbitrary number as the expression simplifies due to orthogonality between the normal and the eigenvectors of the Weingarten operator. Metric intersection [17] allows to generate a final metric which takes into account both requirements.

3 Surface mesh generation

In this section, a comparison between a pure two dimensional and a three dimensional mesh generator is conducted first. Then, the main steps of the meshing strategy advocated in this paper are recalled. It is assumed that the boundary edges have been unfolded in the parametric domain and a valid two dimensional mesh has been obtained.

3.1 Comparison between pure parametric and three dimensional meshing.

During the implementation of the present mesh generator, numerous difficulties have been met when geometry is seen through a lumped metric evaluation:

- When a sizing field is provided to the mesh generator, it is expected that the final three dimensional straight mesh edges will conform to this size distribution. However, the two dimensional mesh generator only evaluates curved edges on the surface as the first fundamental form is used to transfer the information. Even though both should be very close for a very fine mesh, practical considerations such as time and memory requirements may create a significant difference for relatively coarse meshes. Assuming that the metric evaluation is infinitely precise, a length on a curve is measured while ultimately the mesh edge length is the relevant quantity.
- As noted in [24], the CAD parametrization is most of the time not uniform. For a very accurate method such as the advancing front, an inaccuracy in the metric evaluation gives rise to the wrong three dimensional size, and added noise in the final three dimensional mesh. Three dimensional informations will require a couple of iterations, which may be worth to avoid a subsequent intense optimization stage.
- The geometric approximation is most of the time embedded in the metric, without explicitly verifying the relevancy of the information on the three dimensional geometry. There is therefore no guarantee that a valid two dimensional mesh based on some kind of appropriate metric will produce a valid three dimensional surface mesh. As noted in [12], the metric guarantees a zero order approximation of the surface while a triangulation requires implicitly a first order approximation, where the discrete tangent plane approximates reasonably well the analytic one.
- The various three dimensional curves representing the boundaries of the surfaces may have been obtained through complex geometrical operations such as boolean, sweeping or filleting. A representation of these curves is then sought in the parametric domain. This operation adds another layer of approximation to the numerical precision of these curves. Therefore, the evaluation of a curve from the parametric domain to the three dimensional domain may give locations noticeably different from the original three dimensional curve.
- Along the lack of guarantee to create optimal three dimensional triangles from two dimensional triangles, the opposite case is extremely annoying for highly non linear but real mappings. The mesh is only a linear representation of the mapping. Therefore the validity of a linear triangle in the plane may hinder the creation of its valid and optimal three dimensional counterpart.
- Necessary technicalities such as periodicities and degeneracies are completely removed in the three dimensional space.

Summing up, it is really not clear even on a pure efficiency basis if two dimensional parametric mesh generators are faster than their three dimensional counterparts, particularly when no quality argument has been taken into account.

3.2 Algorithm

Once a mesh respecting the boundary has been obtained, the next step is to generate an appropriate surface mesh for analysis. This task is subdivided into two subtasks. The first part generates a mesh in the parametric domain with pure two dimensional information. Edges are sorted in a binary tree and the longest and shortest edge are extracted at each iteration. These edges are possible candidates for split and collapse. Local operators are illustrated in Figure 1. The second part generates a high quality mesh through an advancing front point placement. Compared to previous methods that couple advancing front with global methods [18, 19], the creation of the element generated by an advancing front method is strictly enforced, as described in [1]. The hybrid advancing front strategy proposed in [1] is used. A valid mesh is available at each iteration as only local operations are applied to the mesh. The main algorithmic steps for a parametric mesh generator read:

As long as the front is not empty:

- Select an edge `(ip1,ip2)` in the front
- Compute the three dimensional size, normal and coordinates at the three dimensional mid point.
- Based on the previous size, iteratively compute a three dimensional point `ipnew` such that the new triangle to be well shaped in three dimensions
- March from each end point of the current edge towards `ipnew` :
  - If there are points of the front too close to the new point, put them in a stack and stop the progression
  - If there are points of the front too close to the newly formed edges `(ip1,ipnew)` and `(ip2,ipnew)`, put them in a stack and stop the progression
  - If the front is topologically crossed, put the local points belonging to the front in a stack and stop the progression
- If the progression has not been stopped, check for close points not belonging to the front that could be used in lieu of `ipnew`.
- Else sort the current points of the stack with respect to the quality of the triangle they form with the new edge. Set `ipnew` to this point and come back to the beginning.
- If `ipnew` is not already in the mesh, insert it by splitting the face containing it
- Recover edges `(ip1,ipnew)` and `(ip2,ipnew)`
- Update the front
- Swap in front of the newly created element

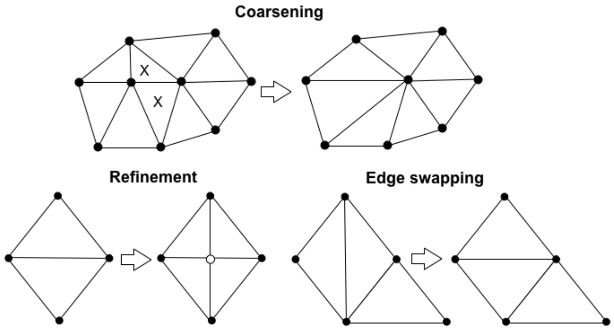

Fig. 1. Illustration of local operators.

Regarding the front strategy, different options are possible. If the three dimensional size is isotropic, then the sorting of the front based on the shortest three dimensional length seems to be an attractive choice, as advocated in [24]. The main reason of using a priority queue for the advancing front strategy is to avoid large elements overlapping small elements [16]. It furthermore allows for a denser point distribution as the small edges will fill less space. However, based on experience, it was found that sorting the front based on layers from the boundaries while locally correcting the size during an iteration process was giving the most regular meshes.

If the three dimensional size is anisotropic, the discussion is more complex, and has not received much attention in the literature. The shortest edge length strategy does not imply in the anisotropic context that the smallest element will be created. It however gives priority to elements with small angles compared to elements with large angles, which are well known to provide better interpolation properties [22, 3]. This deserves further investigation.

4 Singularities

As commented before, geometric degeneracies appear in a NURBS context when control points are repeated. This may be useful for different reasons:

- Create simple shapes such as a cone, a sphere or part of it with NURBS.
- Remove from the basic four sided NURBS patch a side to obtain a three sided surface in the three dimensional space.
- Create discontinuities in the geometric model, either on the boundaries or inside. For example, trailing edges of wings may be created without topological entities by defining discontinuities inside the geometrical surface.
- Create complex shapes with few surfaces in order to avoid intersection computations and approximate reparametrization of curves into the parametric plane.

- Create complex shapes with few surfaces in order to drive a shape optimization and being able to continuously deform the CAD surface.

From a meshing viewpoint, the downside of using repeated control points arises when derivatives of the parametric mapping are to be asked. Repeated control points give null derivative vectors in the direction of the repetition. Therefore, the tangent plane, the normal vector, or the principal curvatures, and all computations relying on derivatives of the parametric mapping are ill defined. As seen in Section 2, the parametric plane transfers the three dimensional sizing information through the use of the tangent plane and the first fundamental form. The singularity of the tangent plane will give rise to null eigenvalues in the metric, and equivalently an infinite size in the parametric domain. Even though numerical cut offs may be used for a safe implementation, a two dimensional mesher becomes blind around these points. Furthermore, this phenomenon only worsens as the mesh is refined, as the singularity is not confined locally, but spreads at least up to the next row of control points. This drawback is solely due to derivatives of the mapping. Therefore, only zeroth order information may be used, as the geometry is nevertheless valid.

From a geometric viewpoint, normal evaluation is extremely important to evaluate the validity of the three dimensional surface mesh. The normal vector is typically normed so that only its direction is relevant. At the opposite, the tangent plane contains directions and parametric mapping informations. We first devised a regularization method which is able to provide relevant normal computations around singularities. This will be describe in the first part. However, this does not remove the degeneracies of the parametric mapping and special techniques have been therefore sought. This is presented in the next part.

4.1 Tangent plane and normal evaluation

In this part, only normal direction is sought. Note that, regarding the normal computation, a unique discrete normal may always exist [2] in the three dimensional space, as long as there is enough visibility around it. Once a normal becomes available, the advancing front procedure may be defined [15, 20]. Based on this remark, a three dimensional evaluation of the normal is proposed to treat theses cases in a uniform approach. Singularities are seldom in the whole patch. Even though a fast detection should be designed to treat them, the cost of the treatment is not really relevant due to the very low probability of hitting a singularity. The idea proposed in this paper is to regularize the parametrization in a discrete and local manner. If a singularity is identified by a collinear or incomplete tangent plane, the normal at the singularity may still be computed relying on relevant neighboring normals in the three dimensional space. The difficulty is due to the fact that the relevant three dimensional values may define a complex parametric bounding box in the parametric plane. Neighboring relevant values may be close in the parametric domain but still the average may require a large extent such as the tip of a cone, as illustrated on Figure 2, in order to obtain a well balanced normal. Therefore, two criteria must be taken into account, namely the non singularity of the tangent plane and the three dimensional proximity. A relevant axis aligned box is hence sought iteratively in the parametric domain around the singular point by iteratively looking for a non singular tangent plane in the u and v direction. A small parametric value is initially set, and is progressively increased if no complete tangent plane is found, or a close three dimensional distance to the singular point is found. As a relevant bounding box is found, a regular grid of points is created and the tangent plane at these points is queried. If they are valid, the normals are computed, accumulated and averaged in an appropriate form [2] to provide the final normal.

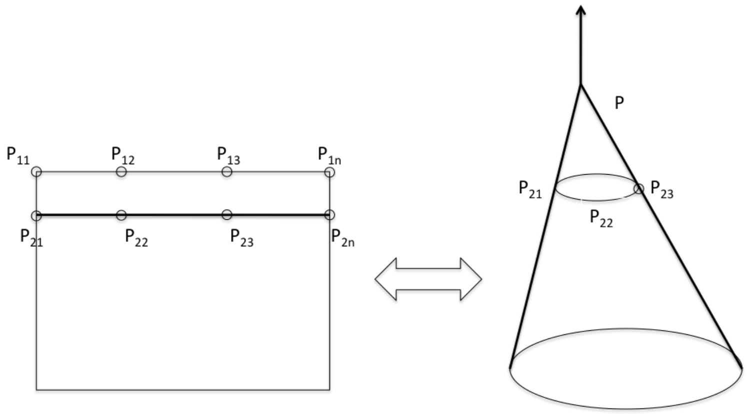

Fig. 2. The normal and tangent plane evaluation is averaged through various points close to the singularity. The first row of points in the parametric domain collapses to the cone tip. Various rows may be used to average the normal.

4.2 Meshing singularities

The main idea here consists in always relying on zeroth order computation, namely point location evaluations in order to avoid the drawbacks evoked before. As noticed in the introduction, some ideas were presented in the literature. They do not however treat the generality of cases met in practice.

In this work, two main steps are involved in meshing around a singularity. The first consists in obtaining well positioned points, the second consists in creating the connectivities between these points. A singular edge in the parametric domain implies that one of the parametric directions is singular at some location and will surround the singularity for values close to the singular parametric values, while the other direction goes into the singularity. In the three dimensional space, this is very close to the geodesic polar map on the surface [11]. The idea consists therefore in meshing layers of elements around the singularity until the mapping becomes reasonable. In the parametric plane however, these layers do not match a u- or v-isoline. In the three dimensional space, edges abut on this singularity. Therefore, the average size of these edges is given for each layer generated.

In order to generate the points, the middle point of the singular edge in the parametric domain is first computed. The direction normal and tangential to the singular edge are computed. This middle point is the initial guess of a zeroth order smoothing where the smoothing is applied independently in both directions. The normal direction allows the current point to be closer or further to the singularity. The tangential direction controls roughly the orthoradial direction close to the singularity only. Beginning from both end points of the edges that abut on the singularity, which may be the same point in the three dimensional space, the point that is at a given distance of the layer from the singularity and at equidistance from the two edge points is sought. If no distance is violated, the point is stored and the same approach is performed recursively. In Figure 3, an octant of a sphere is represented with a singular pole. Relying on the first two edges AB_1 and AC_1 an average size is computed . The parametric edge B_1C_1 is split and the point equidistant from B_1 and C_1 at the correct distance from the singularity is sought. The same procedure is repeated for both edges B_1B_2 and C_1C_2. The main difficulty comes from the fact that the mapping is highly distorted. A small change in the parametric plane does not produce a small change in the three dimensional space. A local scale in the three dimensional space should be first found and all candidates points are created with this local length. The smoothing in the direction towards the singularity is driven by the three dimensional distance to the singularity. The distance in the direction tangential to the singularity is driven by the distance to the plane equidistant to the two ancestor points.

The connection of these points relies on the hierarchical character of the previous point creation. A point is created from two ancestors and hence receives a level number in this process corresponding to the depth of the recurrence. For the creation of each new stripe, a stack of new points is created. The goal is therefore to find the best way to connect these two stripes of points. For each two stripes of points, both stripes are scanned concurrently until a match is obtained on the recurrence level. Once a match has been found, each substripe is meshed with a local one dimensional advancing front procedure as it boils down to choosing a diagonal to connect both sub stripes. Finally, an optimization on each layer is performed in order to obtain the smoothest geometry.

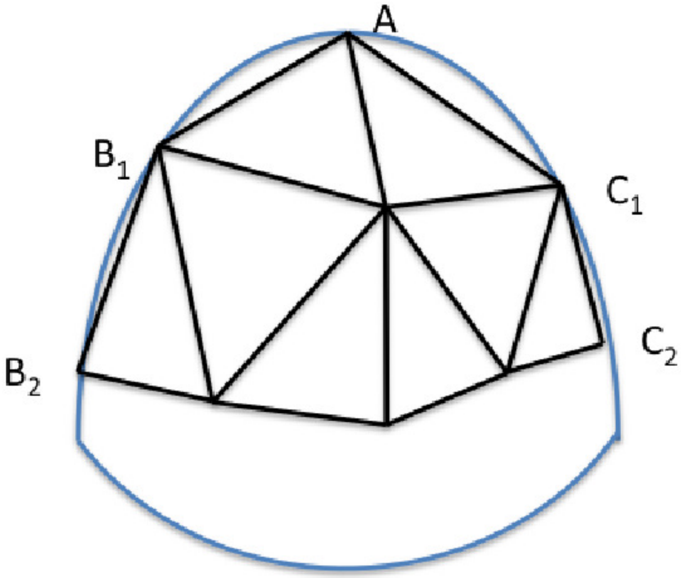

Fig. 3. A degenerate pole of an octant of a sphere.

5 Numerical examples

In this section, numerical examples are provided to illustrate the quality and robustness of the method.

5.1 Cone

This example may certainly be the simplest example with a singularity. Figure 4 shows the mesh of the cone. It consists in $48 \cdot 10^3$ triangles and $24 \cdot 10^3$ points. A zoom close to the tip is shown in Figure 5. For this particular example, only one single layer of elements has been created, as the degeneracy was not judged sufficient to keep on adding layers.

By refining the mesh close to the singularity, it affects more the meshing process as shown in Figure 6. This time, around 15 layers are necessary to get away from the singularity, as shown in Figure 6.

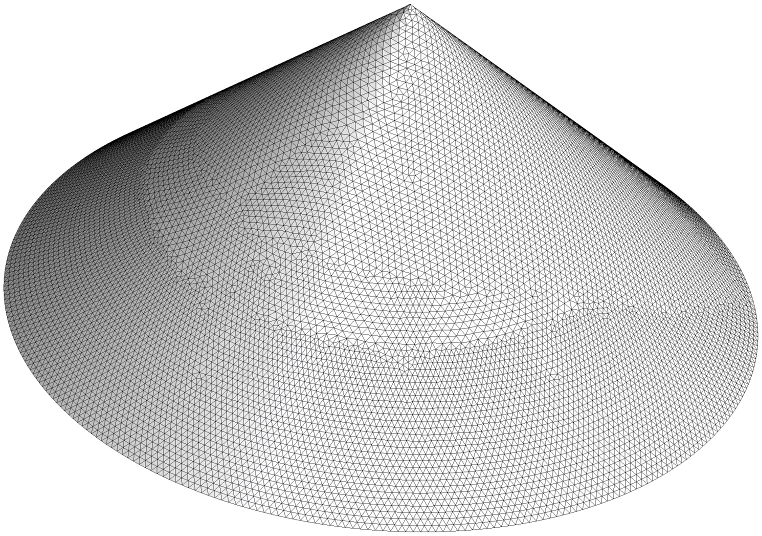

Fig. 4. Cone example.

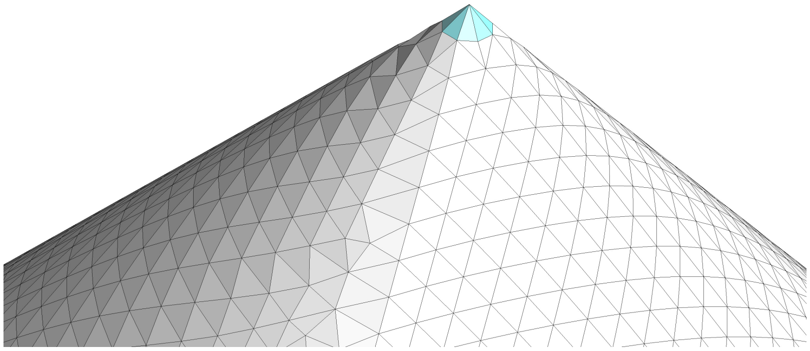

Fig. 5. Zoom on the cone tip.

5.2 Propeller wing

For this example, a highly twisted propeller wing has been chosen. The topology of the wing is complicated as only the trailing edge and the edge from the junction with the body are present. The wing surface is therefore completely wrapped over the trailing edge. Furthermore, a singular point arises at the tip of the wing. The mesh is displayed in Figure 8. The mesh contains $24 \cdot 10^3$ points and $73 \cdot 10^3$ triangles and has been refined with regard to curvature. A zoom close to the tip of the wing is displayed on Figure 9 and another zoom shows the singularity on Figure 10. It is clearly seen that no artifacts in the mesh appear, and a high quality mesh has been obtained close to the singular

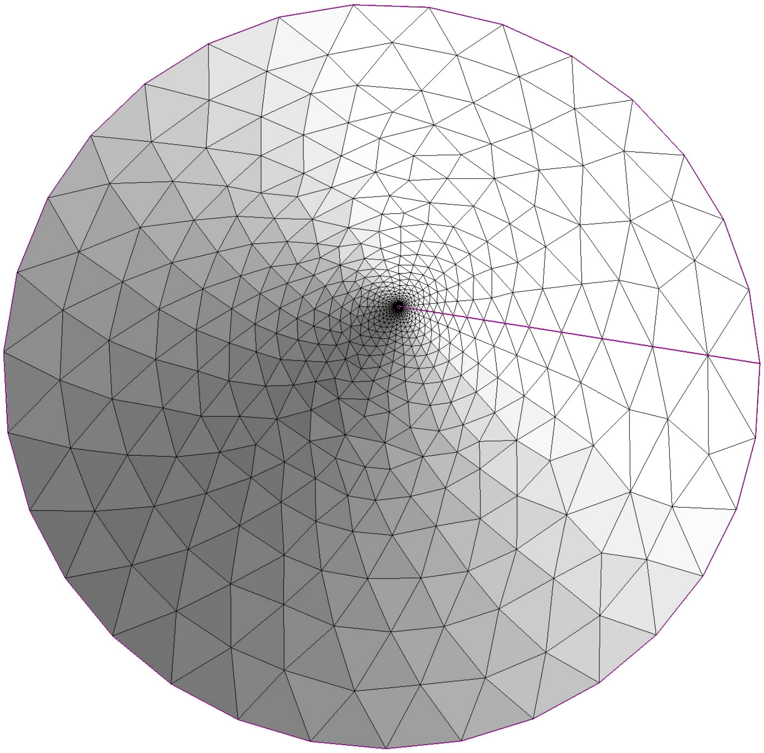

Fig. 6. Refined cone.

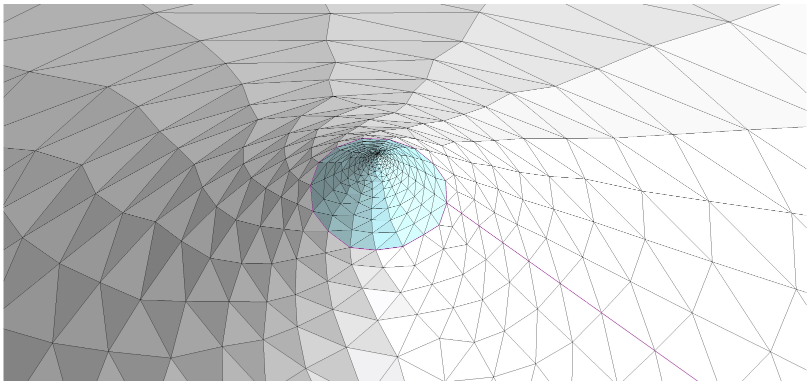

Fig. 7. Zoom on the refined cone tip.

point location. In this example, five layers have been needed to get away from the singularity.

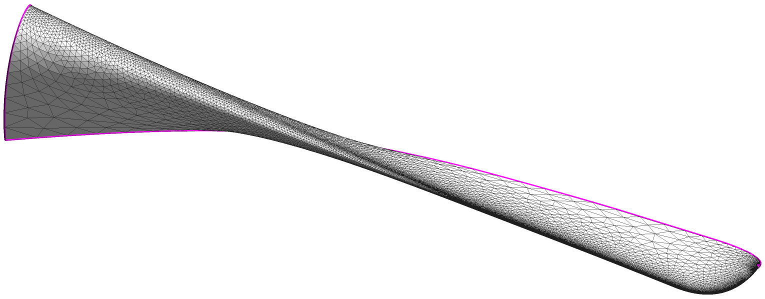

Fig. 8. Mesh for a propeller wing.

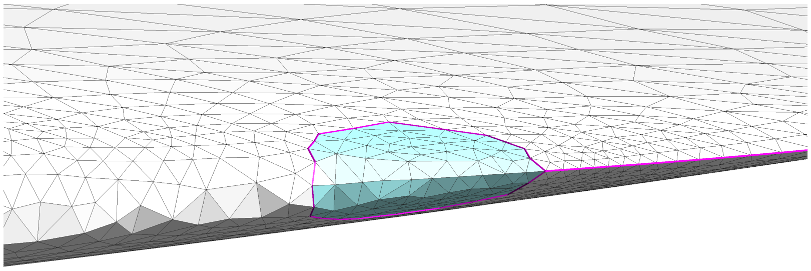

Fig. 9. Zoom close to the tip of the propeller wing.

5.3 Generic fighter

This example illustrates the geometry of a generic fighter. It is composed of 14 faces. The wing tips on the back are composed with one NURBS surface each. The mesh is composed of $68 \cdot 10^3$ points and $137 \cdot 10^3$ triangles. The wing and stabilizer of the fighter are displayed in Figures 12 and 13. Each of the wings and stabilizers have a singular point. Again, singularities have been handled properly.

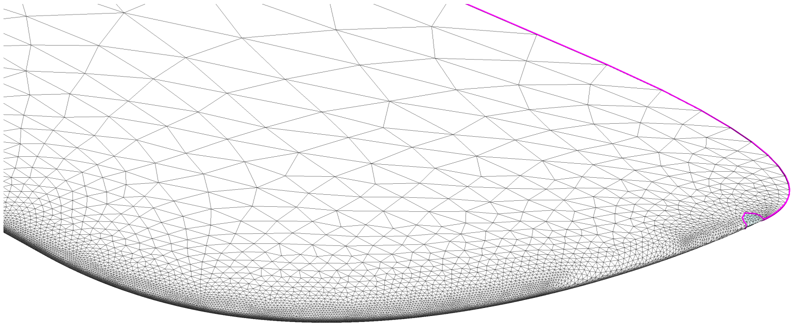

Fig. 10. Tip of the singularity on the propeller wing.

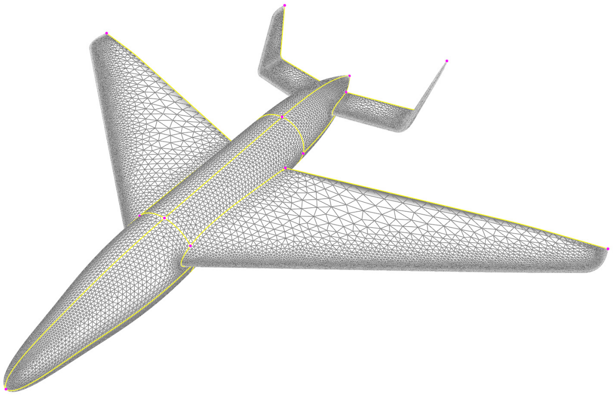

Fig. 11. Mesh for a generic fighter.

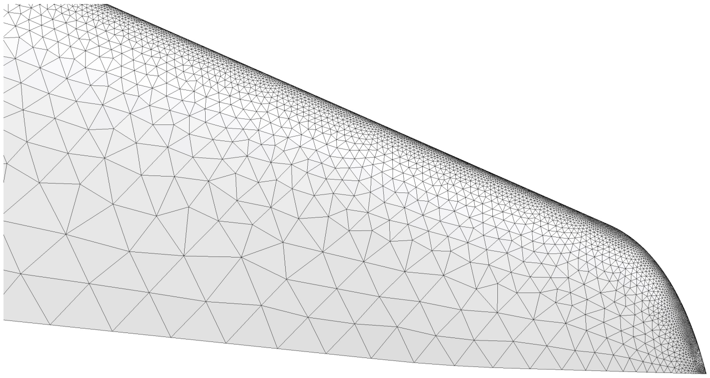

Fig. 12. Close up of the wing fighter near the wing tip.

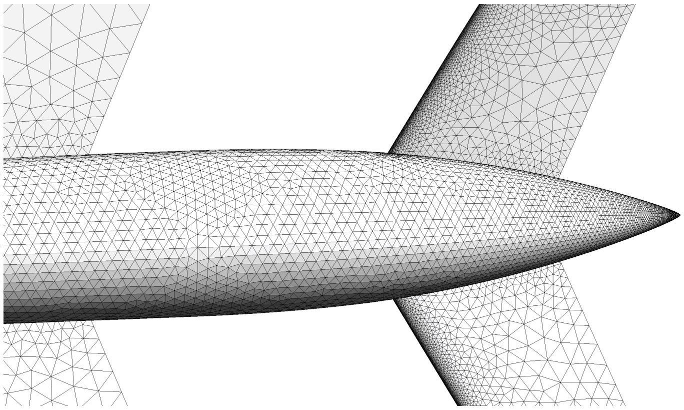

Fig. 13. Close up of the wing fighter near the back of the fighter.

6 Conclusion

A new parametric meshing technique has been presented. Parts of the mesh generation have been highlighted. Particular emphasis has been given to the treatment of singularities in the parametric mapping, which cause havoc to parametric meshers. The main idea consists in relying only on zeroth order information and creating semi structured layers of meshes around the singularities. Examples have been shown that illustrate the quality and robustness of the method.

Acknowledgments

This work was partly supported by the DoD HPCMP CREATE Program. We would like to thank John Livingston from the CREATE AV-DaVinci team for providing the geometries for the twisted wing and the fighter.

References

1. R. Aubry, G. Houzeaux, and M. Vázquez. A surface remeshing approach. *Int. J. Num. Meth. Eng.*, 85(12):1475–1498, 2011.
2. R. Aubry and R Löhner. On the 'most normal' normal. *Comm. Num. Meth. Eng.*, 24(12):1641–1652, 2008.
3. I. Babuska and A.K. Aziz. On the angle condition in the finite element method. *SIAM J. Num. Anal.*, 13:214–226, 1976.
4. T. Banchoff and S. Lovett. *Differential geometry of curves and surfaces.* Natick, MA, A.K. Peters, 2010.
5. H. Borouchaki and P. L. George. Maillage de surfaces parametriques. partie I aspects théoriques. *C. R. Acad. Sci.*, 7(34):833–837, 1997.

6. H. Borouchaki and P. Laug. Simplification of composite parametric surface meshes. *Eng. Comput. (Lond.)*, 20(3):176–183, 2004.
7. H. Borouchaki, P. Laug, and P. L. George. Parametric surface meshing using a combined advancing-front generalized Delaunay approach. *Int. J. Num. Meth. Eng.*, 49:233–259, 2000.
8. J. C. Cuillière. An adaptive method for the automatic triangulation of 3D parametric surfaces. *Computer-Aided Design*, 30(2):139–149, 1998.
9. P.L. George and H. Borouchaki. *Triangulation de Delaunay et maillage.* Hermes, Paris, 1998.
10. Z. Guan, J. Shan, Y. Zheng, and Y. Gu. An extended advancing front technique for closed surfaces mesh generation. *Int. J. Num. Meth.*, 74(1):642–667, 2007.
11. W. Kühnel. *Differential Geometry, Curves – Surfaces – Manifolds.* AMS Student Mathematical Library Series Vol. 16, American Math. Society, 2nd edition 2006, 1998.
12. P. Laug. Some aspects of parametric surface meshing. *Finite Element in Analysis and Design.*, 46:216–226, 2010.
13. C.K. Lee. Automatic metric advancing front triangulation over curved surfaces. *Eng. Comp.*, 17(1):48–74, 2000.
14. C.K. Lee and R.E. Hobbs. Automatic adaptive finite element mesh generation over rational b-spline surfaces. *Comput. Struct.*, 69:577–608, 1998.
15. R. Löhner. Regridding surface triangulations. *J. Comput. Phys.*, 126:1–10, 1996.
16. R. Löhner. *Applied Computational Fluid Dynamics Techniques: An Introduction Based on Finite Element Methods.* Wiley, 2008. Second edition.
17. A. Loseille and R. Löhner. On 3D anisotropic local remeshing for surface, volume, and boundary layers. In *Proc. in 18th International Meshing Roundtable*, pages 611–630, 2009.
18. D.L. Marcum and N.P. Weatherill. Unstructured grid generation using iterative point insertion and local reconnection. *AIAA J.*, 33(9):619–1625, 1995.
19. M.L. Merriam. An efficient advancing front algorithm for Delaunay triangulation. *AIAA-1991-792*, 1995.
20. K. Nakahashi and D. Sharov. Direct surface triangulation using the advancing front method. *AIAA-95-1686-CP*, 1995.
21. J. Peraire, J. Peiro, and K. Morgan. Adaptive remeshing for three-dimensional compressible flow computations. *J. Comput. Phys.*, 103:269–285, 1992.
22. S. Rippa. Long and thin triangles can be good for linear interpolation. *SIAM J. Numer. Anal.*, 29(1):257–270, 1992.
23. D. Rypl and P. Krysl. Triangulation of 3D surfaces. *Engineering with Computers*, 13:87–98, 1997.
24. J.R. Tristano, S.J. Owen, and S.A. Canann. Advancing front surface mesh generation in parametric space using a Riemannian surface definition. In *IMR*, pages 429–445, 1998.